

MÉTODOS DE MÉTRICAS ORIENTADOS A LOS PUNTOS DE FUNCIÓN

ORIENTED METHODS OF METRIC TO THE FUNCTION POINTS



AUTOR

ALBEIRO CUESTA MEZA
Ingeniero de Sistemas
Especialista en Administración de Sistemas
Doctorando Ingeniería del Software
Investigador Universidad Autónoma de Manizales
albeirocuesta@autonoma.edu.co
COLOMBIA.

AUTOR

MARCELO LÓPEZ TRUJILLO
Ingeniero de Sistemas
Especialista en Administración de Sistemas
Magíster en Educación
Doctorando Sociedad de la Información y del conocimiento
Profesor Asociado Universidad de Caldas
mlopez@ucaldas.edu.co
COLOMBIA.

AUTOR

LUÍS JOYANES AGUILAR
Doctor Ingeniería Informática
Doctor
Catedrático Universidad Pontificia de Salamanca
joyanes@gmail.com
ESPAÑA

INSTITUCIÓN

Universidad Autónoma de Manizales
UAM
Carrera 9A No. 19-03
www.mngr@umanizales.edu.co
Teléfono: (6) 88441450 ext 296
CALDAS, MANIZALES - COLOMBIA

INSTITUCIÓN

Universidad de Caldas
Calle 65 No. 26 - 10
ucaldas@ucaldas.edu.co
Teléfono: (6) 8781500
CALDAS, MANIZALES - COLOMBIA

INSTITUCIÓN

Universidad Pontificia de Salamanca UPSA
C/compañía, 5-37002
webmaster@upsa.es
Teléfono: +34923277100
SALAMANCA, ESPAÑA

Recepción: Septiembre 8 de 2009

Aceptación: Agosto 16 de 2009

Temática: Nuevas Herramientas y Tencologias de Desarrollo Software

Tipo de Artículo: Reflexión

RESUMEN

La importancia adquirida por los puntos de función¹ se puede constatar tanto en la enorme difusión de su aplicación, como en la diversidad de los métodos desarrollados, o en la estandarización de dichos métodos. Este último aspecto, la generación de esta diversidad de métodos de medición, ha introducido fundamentalmente dos factores de dificultad a la hora de utilizar puntos de función, como son, por una parte, la comparación de medidas realizadas con diferentes métodos, y por otra la elección del método más adecuado para el usuario y el proyecto. Este artículo presenta cuatro de los métodos utilizados a nivel mundial, FPA, COSMIC-FFP, IFPUG y MKII, difundiendo las características más relevantes de cada uno de ellos.

1. Los Puntos Función derivan un conjunto de métricas esenciales para la gestión de la productividad, la calidad y el costo del software.

PALABRAS CLAVES

Puntos de función,
Métricas,
FPA,
COSMIC-FFP,
MKII,
IFPUG,
Calidad del software.

ABSTRACT

The importance acquired by the function points can be verified so much in the enormous diffusion of their application, like in the diversity of the developed methods, or in the standardization of these methods. This last aspect, the generation of this diversity of measurement methods, it has introduced two factors of difficulty fundamentally when using function points, such as, on one hand, the comparison of measures realized with different methods, and for other, the election of the most appropriate method for the user and the project. This article presents four of the methods used at world level, FPA, COSMIC-FFP, IFPUG and MKII, diffusing the most outstanding characteristics of each one of them.

KEYWORDS

Function points,
Metrics,
FPA,
COSMIC-FFP,
MKII,
IFPUG,
Software quality.

INTRODUCCIÓN

La medición del software es actualmente un medio esencial para realizar las estimaciones oportunas de diversos indicadores necesarios para el desarrollo de productos software. De este modo, la precisión de los instrumentos de medición juega un rol importante porque se constituyen en herramientas de soporte para los administradores de proyectos de software.

Un instrumento de medición no es más que la automatización de un procedimiento de medición¹ previamente definido.

¹ El Vocabulario Internacional de Términos Básicos y Generales en Metrología define a un Procedimiento de Medición como un conjunto de operaciones descritas específicamente y usadas en la ejecución de una medición particular de acuerdo a un método dado.

De manera general, los métodos de estimación han sido diseñados para medir un determinado tipo de software. Por tanto, la aplicación de cada método depende particularmente del dominio software o del tipo de desarrollo. Existen por un lado, los sistemas tiempo real, científicos, ingeniería, entre otros, donde no sólo se deben tomar en cuenta los parámetros convencionales del software sino también parámetros adicionales propios a cada tipo. Por otro lado, existen los sistemas basados en un análisis y diseño orientados a objetos, cuya medición depende mucho de las características de ese tipo de software así como en su caso, de la utilización de la notación UML en todas las etapas de desarrollo.

Por otra parte, la publicación de la norma ISO de medición de tamaño funcional ha permitido la definición de reglas importantes con el objetivo de realizar mediciones correctas únicamente de la funcionalidad del sistema. Esta norma ha influido considerablemente sobre la creación de métodos de medición como COSMIC-FFP, permitiendo así realizar estimaciones simplificadas y genéricas aplicables a un gran número de tipos de software.

El presente artículo está organizado de la siguiente manera: en la sección 2 se mencionan las características mas relevantes de los métodos de métricas orientados a los puntos de función y una aplicación de cada método con el mismo ejemplo y en la sección 3 se dan las conclusiones.

1. CARACTERÍSTICAS DE LOS MÉTODOS DE MÉTRICAS ORIENTADOS A LOS PUNTOS DE FUNCIÓN

En esta sección, se presentan las características más relevantes de cada uno de los métodos y una comparación acerca de los mismos:

1.1 METODO MKII

Los puntos de función MkII - FPA se obtienen sobre el producto de dos componentes: el tamaño de procesamiento de la información y el ajuste de complejidad técnica.

En el Tamaño del Procesamiento de la Información en vez de utilizar cinco componentes: entradas, salidas, interfaces, consultas y ficheros lógicos, el sistema se examina desde el punto de vista de las transacciones lógicas, consistiendo cada una de ellas en entrada, proceso y salida.

El ajuste de complejidad técnica se consideran los siguientes factores: Ampliar la lista de características generales. Se incluyen 5 características específicas mas y se permite introducir otras.

1.2 MÉTODO FPA

Es un método para medir el tamaño funcional de un sistema de información. FPA mide el tamaño funcional mirando las transacciones (funcionales) y los ficheros de datos (lógicos) que son relevantes al usuario en el negocio.

El FPA evalúa con fiabilidad:

- El valor comercial de un sistema para el usuario.
- Tamaño del proyecto, costo y tiempo de desarrollo.
- Calidad y productividad del programador MIS.
- Esfuerzo de adaptación, modificación y mantenimiento.
- Posibilidad de desarrollo propio.
- Beneficios de implementación en 4GL.

1.3 MÉTODO IFPUG

El objetivo principal de este método es medir el software cuantificando la funcionalidad que suministra al usuario basándose principalmente en el diseño lógico. Bajo esa perspectiva, los resultados que se quieren obtener del análisis de los Puntos de Función son:

- Medir la funcionalidad que el usuario requiere y recibe.
- Medir el desarrollo y el mantenimiento del software independientemente del de la tecnología utilizada para su implementación.

Es importante resaltar que el proceso de contar los Puntos Función deberá ser lo suficientemente simple para minimizar el sobre-esfuerzo necesario del proceso de medir.

1.4 MÉTODO COSMIC-FFP

COSMIC-FFP se creó para contar con métodos más precisos para estimación y medición de las características del software que sean igualmente fiables para todos los tipos del mismo, bajo las siguientes necesidades:

1. Conocer la talla funcional del software como componente importante de la medición y de las características que se necesiten.
2. Estimar los requerimientos en la talla del software, como paso a calcular costos del proyecto.

1.5 COMPARACION DE LOS METODOS: SIMILITUDES Y DIFERENCIAS

Los métodos de FPA intentan cuantificar los requerimientos funcionales y no-funcionales. En ambos métodos esto se logra mediante la determinación inicial

del tamaño funcional y la posterior aplicación de un factor de ajuste al tamaño, para atender el esfuerzo adicional necesario para cumplir los requerimientos de calidad. Las principales diferencias entre IFPUG y MKII FPA se encuentran en la forma que se determina el tamaño funcional.

En cualquier método de medición del tamaño funcional se miden los requerimientos funcionales de una aplicación en términos de uno o más tipos de componentes lógicos, cada uno de los cuales debe tener una relación explícita con el límite de la aplicación.

Además, se proveen reglas para identificar los tipos de componentes lógicos del software y asignarles un valor numérico que representa su contribución al tamaño funcional. Después de identificar los componentes, se determina el tamaño funcional de la aplicación mediante la suma del tamaño de cada uno de los tipos de componente lógico.

EI: Entradas Externas.

EO: Salidas Externas.

EQ: Consultas Externas.

ILF: Archivo Lógico Interno.

EIF: Archivo de Interfaz Externo.

FTR: Cantidad de Archivos Referenciados.

DET: Cantidad de Tipos Elementos Dato.

RET: subgrupo de datos reconocible por el usuario dentro de un ILF y EIF.

MKII y IFPUG expresan los requerimientos funcionales en términos de componentes lógicos. MKII usa un único tipo de componente lógico y expresa todos los requerimientos funcionales como transacciones lógicas. IFPUG usa cinco tipos de componentes lógicos: EI, EO, EQ, ILF, EIF.

Los tipos de componentes lógicos están formados por varias partes. En MKII cada transacción lógica se compone de entrada, procesamiento y salida. En IFPUG, EI, EO, se compone de mensajes de entrada y salida, respectivamente, EQ por un par de entrada/salida, ILF y EIF por datos persistentes mantenidos por la aplicación u otra aplicación, respectivamente.

En MKII, cada tipo de entidad se trata como independiente y se cuentan las referencias a las entidades por cada transacción lógica. En IFPUG, los tipos de entidades se agrupan para formar ILF si están dentro del límite de la aplicación o EIF si están fuera del límite. Las referencias a los tipos de entidad se cuentan como FTR por cada EI, EO o EQ.

En ambos métodos hay una relación entre los tipos de componentes lógicos y el límite de la aplicación.

En MKII, por cada transacción lógica, el mensaje de entrada y de salida atraviesa el límite cuando llega y sale de la aplicación respectivamente y la parte de procesos esta completamente retenida dentro del límite. En IFPUG, cada EI y EO atraviesa el límite cuando llega y sale de la aplicación, por cada EQ la parte de entrada y salida atraviesa el límite cuando llega y sale y los ILF Y EIF están completamente retenidos dentro del límite de la aplicación que se analiza o de otra aplicación respectivamente.

Ambos métodos basan sus reglas para evaluar el tamaño de sus componentes lógicos en las cuentas base. En ambos se cuenta la cantidad de DET en las partes del mensaje del componente lógico y la cantidad de referencias a datos persistentes dentro de la aplicación. Aunque usan diferente terminología, las definiciones son similares.

Mientras en MKII las cuentas resultan de las entradas y salidas de las transacciones y de las referencias a los datos persistentes, en IFPUG se identifican ILF y EIF y se cuentan sus partes constituyentes que contribuyen a la funcionalidad.

En MKII se cuenta una referencia a una entidad por cada tipo de entidad accedida durante el curso de una transacción lógica, mientras que en IFPUG se cuenta una FTR por cada ILF o EIF accedido durante el curso de una entrada, salida o consulta.

Los objetos de la especificación que se deben identificar son los mismos para ambos métodos: mensajes de entrada y salida, mensajes de error y tipos de entidad. Además en IFPUG se consideran los DET almacenados en los tipos de entidad.

Las principales diferencias entre los dos métodos surgen desde cómo se construyen las cuentas base y no desde qué es lo que se cuenta.

Para determinar las cuentas base, se identifican los respectivos componentes lógicos y se cuenta la cantidad de DET en los mensajes asociados. Los mensajes de error se tratan de manera diferente en los dos métodos. En IFPUG los atributos de un mensaje de error se tratan como atributos de un mensaje de entrada, en MKII se tratan como atributos adicionales de un mensaje de salida. A continuación se cuenta la cantidad de accesos a datos persistentes. En IFPUG, además, se deben ejecutar pasos adicionales para contar la cantidad de tipos de entidad en los grupos de tipos de entidad que forman cada ILF y EIF y para contar la cantidad DET almacenados en los tipos de entidad.

Hay algunas otras variaciones. MKII usa el concepto de entidad System para agrupar las entidades no primarias que pueden contener información dependiente de la implementación. IFPUG no cuenta datos dependientes de la implementación.

En MKII las cuentas bases se usan directamente en los cálculos del tamaño funcional expresado en FP sin ajustar. En IFPUG las cuentas bases se usan para determinar la magnitud de cada componente lógico. Usando las tablas provistas por el método se valora la magnitud de cada componente como bajo, medio o alto, en base a los valores de las cuentas base de DET y referencias a tipos de entidad o la cantidad de RET en el caso de ILF y EIF. IFPUG usa el término complejidad en vez de magnitud. No debe confundirse con la complejidad técnica.

Los mensajes de entrada, salida y referencias a datos persistentes tienen su propia contribución a una aplicación, pero la contribución de cada una es de diferente clase. Para combinarlas y derivar un único valor numérico del Índice de tamaño funcional, se deben normalizar las cuentas base para usar una única unidad. Esto se logra a través de los pesos relativos.

En MKII, se usan tres pesos: Wi, Wo, We, en IFPUG, el sistema de pesos es mas complicado, debido a la gran cantidad de tipos de componentes lógicos. Cada tipo de componente lógico tiene asignado un peso que depende del tipo de componente y su magnitud.

En ambos métodos el Índice de tamaño funcional de la aplicación expresado en FP se obtiene a partir de la suma de las cuentas afectadas por los pesos.

Los dos métodos usan enfoques similares para considerar los requerimientos técnicos. Se evalúa una lista de características no-funcionales en una escala de cero a cinco. IFPUG usa 14 características; MKII usa las mismas 14, pero agrega otras cinco o más.

Una vez evaluado el grado de influencia de las características, se suman para el grado de influencia total. Este valor se usa para ajustar el Índice de tamaño funcional y obtener un nuevo valor que representa el tamaño funcional y los requerimientos técnicos combinados.

El método COSMI-FFP es totalmente diferente a los otros métodos, COSMIC considera la medición de la talla funcional del software a través de dos fases distintas: la representación del software a ser medido y la medición de los aspectos específicos del modelo.

El modelo de contexto presenta tres elementos principales. Capas, frontera y usuarios del software.

La fase de representación es donde se identifican las capas, fronteras, procesos funcionales, grupos de datos y los atributos de los datos.

- Cada grupo de datos identificado es registrado en una columna.
- Cada proceso funcional es registrado en una línea específica y agrupados por capas identificadas.

La fase de medición se basa en la identificación, cuantificación de subprocesos aplicación de la función de medición y agregación de los resultados.

- Para cada proceso funcional identificado, los subprocesos identificados son anotados en la celda correspondiente utilizando la siguiente convención: "E" por una entrada, "X" por una salida, "R" por una lectura y "W" por una escritura.
- Para cada proceso funcional identificado, los subprocesos son sumados por tipo, y cada sub-total es registrado en la columna apropiada, al lado derecho de la matriz.
- El resumen de la medición puede entonces ser calculado y registrado en las celdas de cada capa, en la línea de los totales.

2. CONCLUSIONES

En este artículo, se han definido los métodos de métricas orientadas a los puntos de función que existen a nivel mundial, mostrando sus principales características de medición para estimar el tamaño funcional de sistemas a partir de especificaciones de requisitos de software. Así mismo, pretendemos también comparar los resultados obtenidos para la selección de un método que se ajuste a las necesidades del usuario final, ya que la medición es una parte esencial del proceso de mejora de cualquier actividad humana. Existe un ciclo de mejora que consta en Medir- Analizar-Tomar Acciones y que debe ser aplicado a procesos, productos y clientes, con el objeto de determinar conceptos como innovación, productividad, costes, calidad del producto, servicio al cliente.

3. REFERENCIAS BIBLIOGRAFICAS

- [1] IFPUG, "Function Point Counting Practices Manual, Release 4.1", International Function Point Users Group, Westerville, Ohio, USA 1999.
- [2] UKSMA, "MKII Function Point Analysis Counting Practices Manual. Version 1.3.1", United Kingdom Software Metrics Association 1998.
- [3] COSMIC-FFP Measurement Manual version 2.2, Common Software Measurement International Consortium, January 2003.
- [4] AEMES. Manual De Medición para los Puntos de Función. Madrid, 2000. 376p.
- [5] DOLADO Cozin, José Javier y Fernández Sanz Luis. Medición para la Gestión en la Ingeniería del Software. 1 Ed. San Sebastián : Ra – Ma, 1999. 270p.
- [6] N. Condori-Fernández, Trabajo de Investigación: Validación Teórica del Procedimiento de Medición de Tamaño Funcional, Departamento de Sistemas Informáticos y Computación, 2004.
- [7] Método de Análisis de Puntos de Función MKII. (2004). [On Line]. Abril de 2004 Reino Unido. Disponible en Internet: <http://www.sc.ehu.es/jiwdocoj/mmis/fpmkii.htm>
- [8] COSMICON. (2003). Measurement manual (The cosmic implementation guide for ISO/IEC 19761: 2003): version 2.2. [on line]. Enero de 2003. Disponible en Internet: http://www-ivs.cs.uni-magdeburg.de/~lother/Teaching/SWQM/Dokumente/CFFP_MM_v2.2e.pdf
- [9] IFPUG. (2003). For Immediate Release : IFPUG Functional Sizing Method published as an International Standard [On Line]. 30 de Noviembre de 2003. Disponible en Internet : <<http://www.ifpug.org/about/ISOPressRelease1203.doc>>
- [10] Albrech, A.: Measuring Application Development Productivity, Proceedings of the /IBM Application Development Symposium, GUIDE/SHARE, Monterrey (California), 1979, pp. 83-92.
- [11] Albrech, A. J., Gaffney, J. E. "Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation". IEEE Transaction on Software Engineering. Vol 9. N° 6. 1983. Pg: 639-648.
- [12] Fenton, N. and Pfleeger, S. Software Metrics. A rigorous and practical approach, PWS., 1997
- [13] International Function Point Users Group (IFPUG). "function Point Counting Practices Manual, Release 4.0". IFPUG. Ohio. 1994.
- [14] IFPUG: Function Points Counting Practices Manual. Release 4.1. IFPUG. Ohio (EE.UU.), 1999.

- [15] United Kingdom Software Metrics Association (UKSMA). MK II Function Point Analysis. Counting Practices Manual. Version 1.3.1
- [16] COSMIC-FFP Manual. Version 2.1. Common Software Measurement International Consortium, 2001.
- [17] International Function Point Users Group (IFPUG). "function Point Counting Practices Manual, Release 4.1". IFPUG. Ohio. 1998.